

Платформенная разработка и топологии команд

Михаил Бижан, ОТП Банк

4.12.2020

Михаил Бижан Chief Technology Officer в ОТП Bank Russia

Руководит функциями:

- Продуктовая разработка
- Enterprise архитектура
- Процессы ИТ
- Инструменты ИТ
- Развитие практик и культуры DevOps



Ранее — Райффайзенбанк, Сбербанк, системные интеграторы

Александр Титов Управляющий партнер Экспресс 42



- Организатор сообщества DevOps Moscow,
- Организатор конференции DevOpsDays Moscow,
- Член программного комитета DevOpsConf Russia

Область профессиональных интересов - соединять технологии и людей, в частности DevOps.

ОТП Банк – универсальный банк в РФ, часть венгерской OTP Group



Масштабы бизнеса в России



12,7 тыс. сотрудников



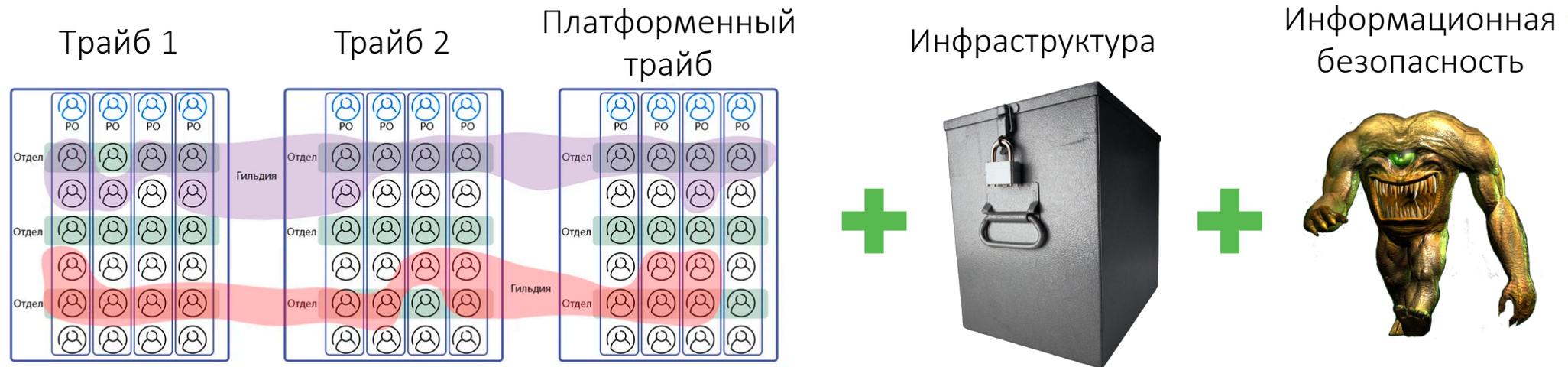
3,7 тыс. населенных пунктов



3,7 млн клиентов



Структура ИТ в половине* российских банков в 2020



* Может и не в половине, это гипербола 😬

Как работает платформенная разработка если тебе не повезло

Ситибанк



Продуктовая команда



Платформенная команда



У нас тоже были платформы



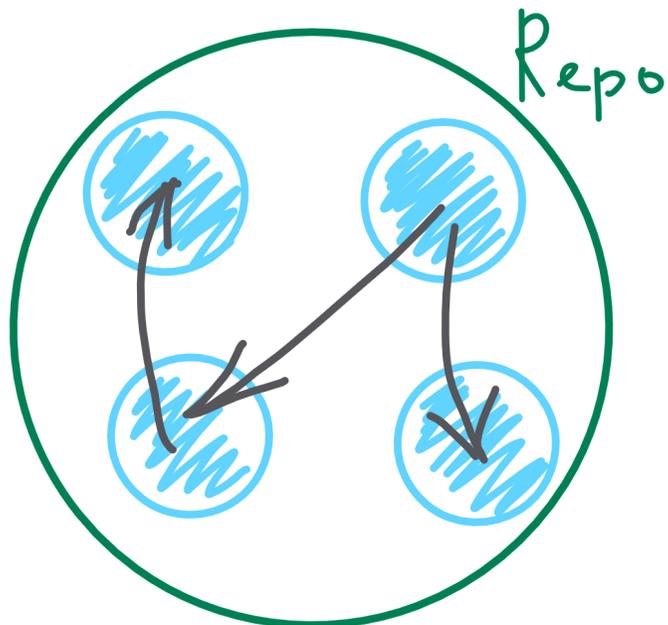
1. АБС (Основная система банка)
2. CRM
3. DWH

~200 чел.

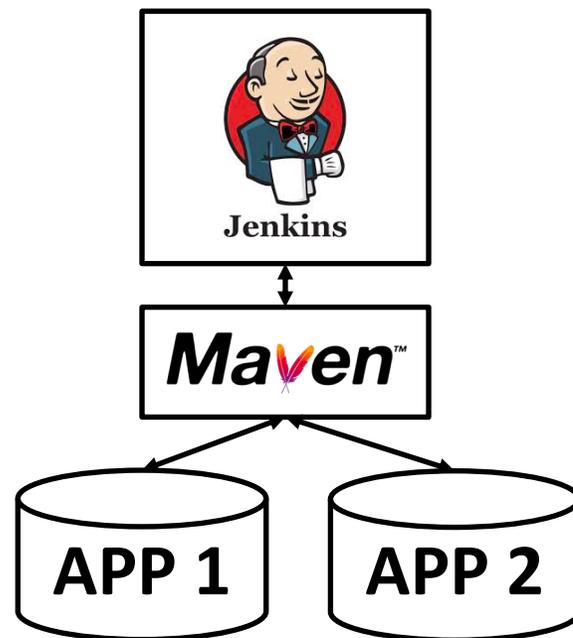
Как они работали? Хорошо, но медленно



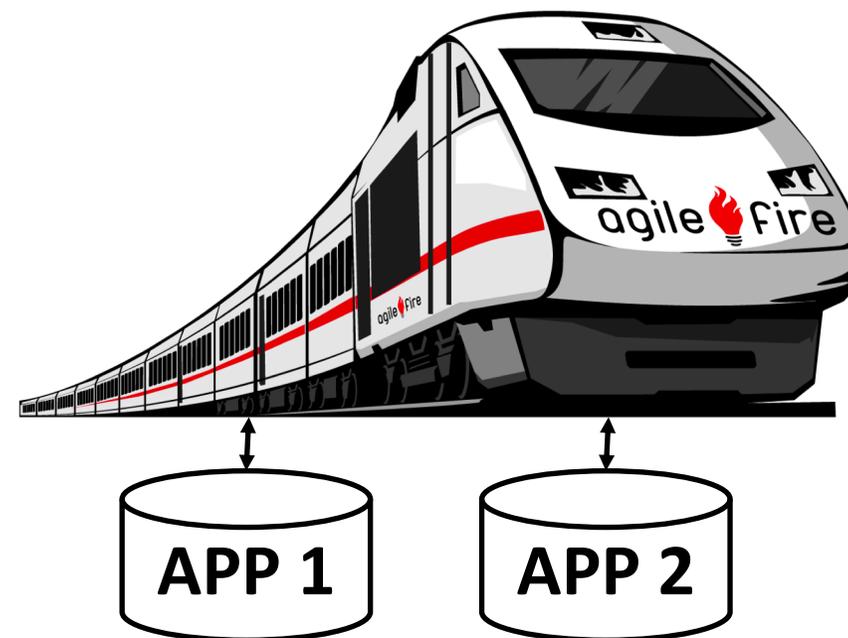
Все old school платформы делают это



Общий репо



Общий Build

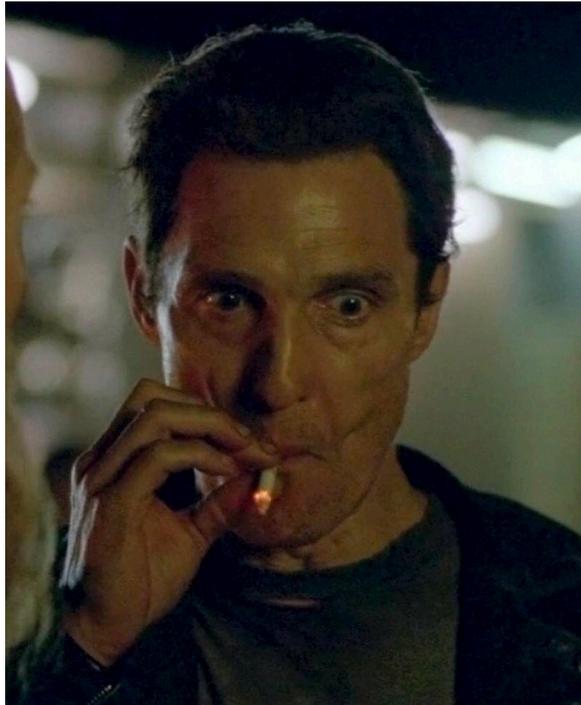


Релиз трейны

Почему платформы никому не нравились.



Техническая сложность



«Да, 100500к зависимостей, а что ты хотел?
Все проекты вместе собираются!» (с)

Current
mood level



Почему платформы никому не нравились.

(Не)предсказуемость



— Когда моя фича появится в платформе?

— Может в мае, может в ноябре...

Current mood level



Почему платформы никому не нравились. (Не)прозрачность



- Почему мою фичу выкинули из релиза?
- Мы провели два груминга и один PBR и решили ее не делать

Current
mood level



Почему платформы никому не нравились.

Обеспечение качества



«Вы платформа — вы и тестируйте» (с)

Current
mood level



**Это длилось годами с периодическими
улучшениями и ухудшениями**

Почему мы вообще решили что-то менять?

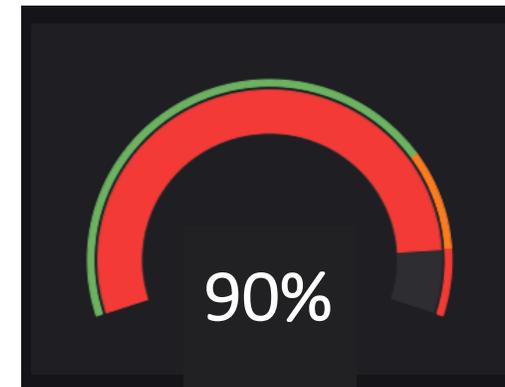


Mean time to recovery



В среднем несколько часов на восстановление при аварии

Change failure rate



9 из 10 деплоев приводят к прерыванию сервиса

Во-вторых, WIP платформ стал расти



CFTACC Kanban v3
Kanban board

QUICK FILTERS: Исп: ГорюновВ Исп: АлехинаЕ Исп: Пушкина Исп: ЧернецоваЕ Исп: КуприковА Исп: М...офееваН Only My Issues Recently Updated

BACKLOG 11 TO DO 6

ANALYSING 31 MAX 8 DEVELOPING 7 MAX 6 TESTING 6 MAX 15 UAT 5 MAX 10 WAITING FOR INSTALL 3 GOLIVE 2 DONE 28 OF 74 Release

Релиз R2-21 3 issues

Релиз R1-21 19 issues

CFTACC-158 Исправление проблемы пролонгации

CFTACC-129 контроль на дублирование отправки

CFTACC-146 Доработка загрузки в ЗП проект КВД и

CFTACC-148 Новый альбом УФЭБС с 1 января 2021 года

CFTACC-154 Подготовка и тестирование дистрибутива R1-

CFTACC-150 Реализация возможности вызова

CFTACC-107 Грузка в ЦФТ росов ЦБ по драг таллам

CFTACC-155 Закрытие не активных счетов, не имеющих

вио-9736 Нехорректное расположение ошибки при

CFTACC-137 Доработка выгрузки данных о кредитах в

CFTACC-166 Оптимизация времени проводки документов ДБО

CFTACC-152 Доработка формирования анкеты клиента

CFTACC-130 Доработка форматов обмена с ПФР [локал

CFTACC-107 Грузка в ЦФТ росов ЦБ по драг таллам

CFTACC-155 Закрытие не активных счетов, не имеющих

вио-9736 Нехорректное расположение ошибки при

CFTACC-137 Доработка выгрузки данных о кредитах в

CFTACC-166 Оптимизация времени проводки документов ДБО

CFTACC-24 Интерфейс удаления операции

CFTACC-105 Доработка кассовых отчетов в связи с



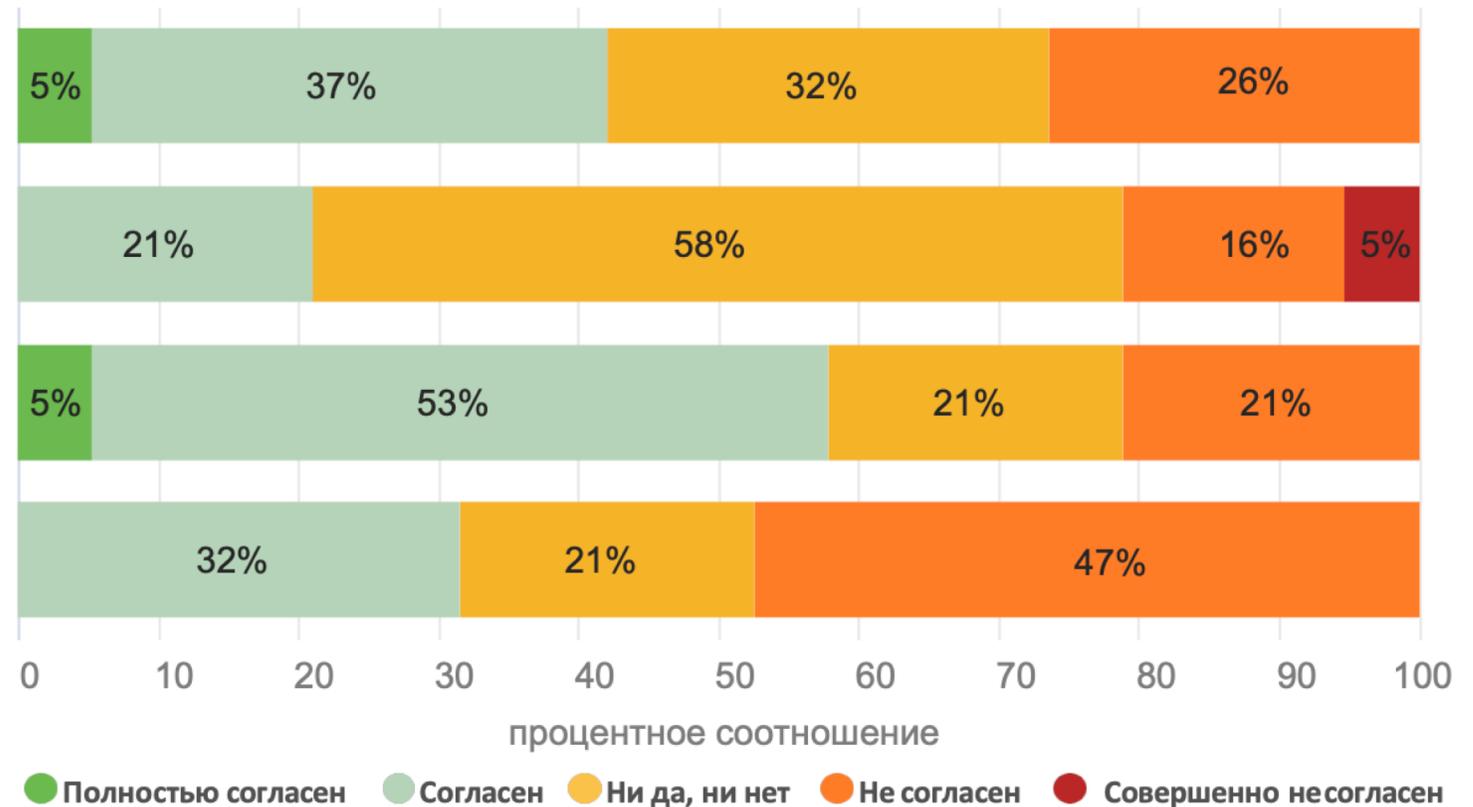
И наконец, у ребят из платформы оказался низкий NPS

5. Мы реагируем на обращения внешних и внутренних клиентов незамедлительно.

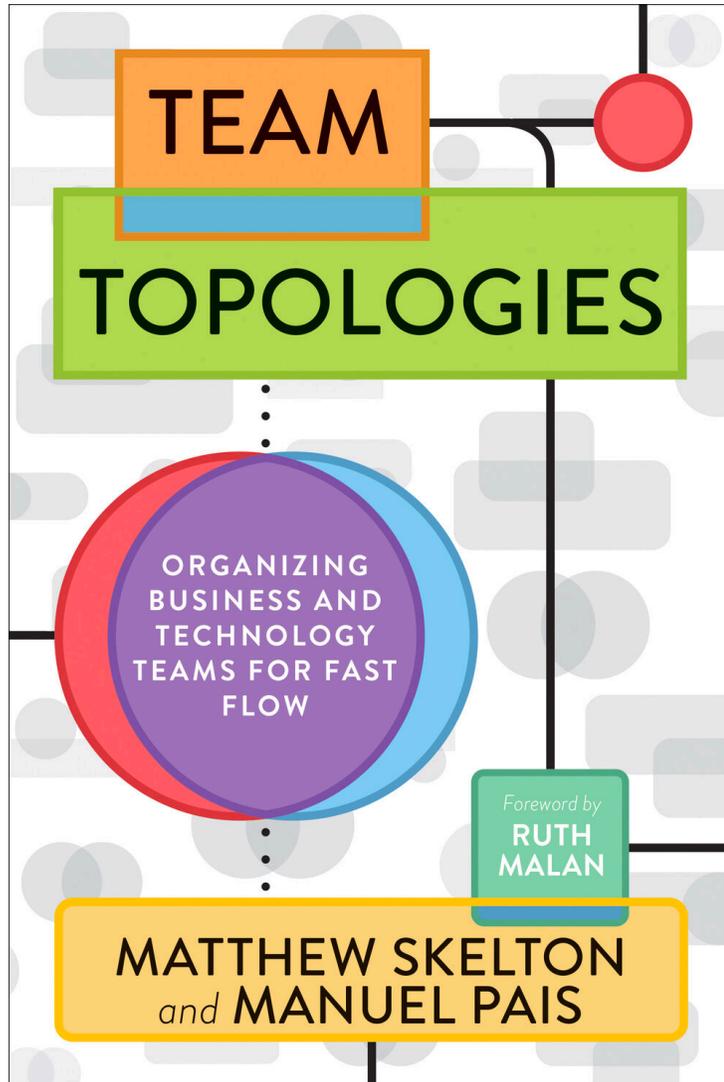
6. Мы разрешаем проблемы внешних и внутренних клиентов незамедлительно.

7. Мы всегда используем обратную связь с внутренними и внешними клиентами для улучшения качества процессов.

8. Как пользователь банковских продуктов ОТП Банка я удовлетворен(а) полностью.



Мы решили, что пора что-то менять



**Monoliths vs
Microservices is
Missing the Point**



The architecture of the system gets cemented in the forms of the teams that develop it.
— Ruth Malan, “Conway’s Law”

Топологии команд



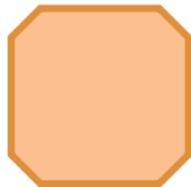
Stream-aligned team

Ориентация на создание ценности



Enabling team

Поддержка



Complicated-subsystem team

Сложные подсистемы



Platform team

Платформа

Топологии команд. Примеры



Stream-aligned team

Создают поток ценности (продукт, сервис)

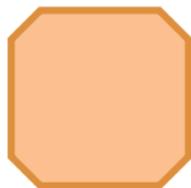
Например, чат на сайте



Enabling team

Эксперты, который «энейблят» другие команды

Например, внедряют практики CI/CD



Complicated-subsystem team

Развивают системы, требующие редкую экспертизу

Например, централизованный кластер Rabbit и нужен Erlang

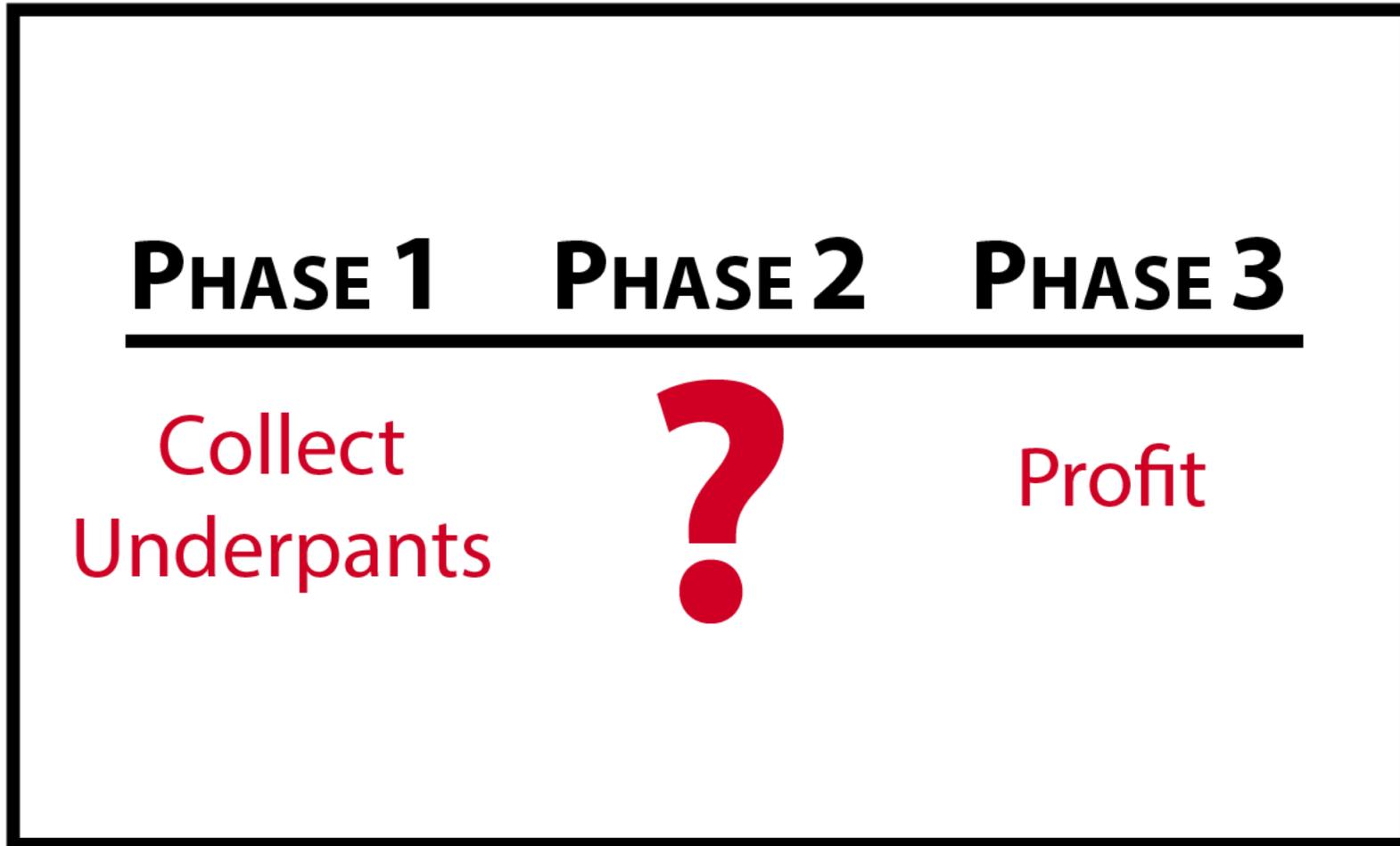


Platform team

Внутренние сервисы для работы других команд

Например, общая CRM система или iOS-приложение

Как же сделать из команды платформу?



Если серьезно

PHASE 1

PHASE 2

PHASE 3

Убрать все
лишнее

Добавить все
недостающее

Profit

Если серьезно и детальнее



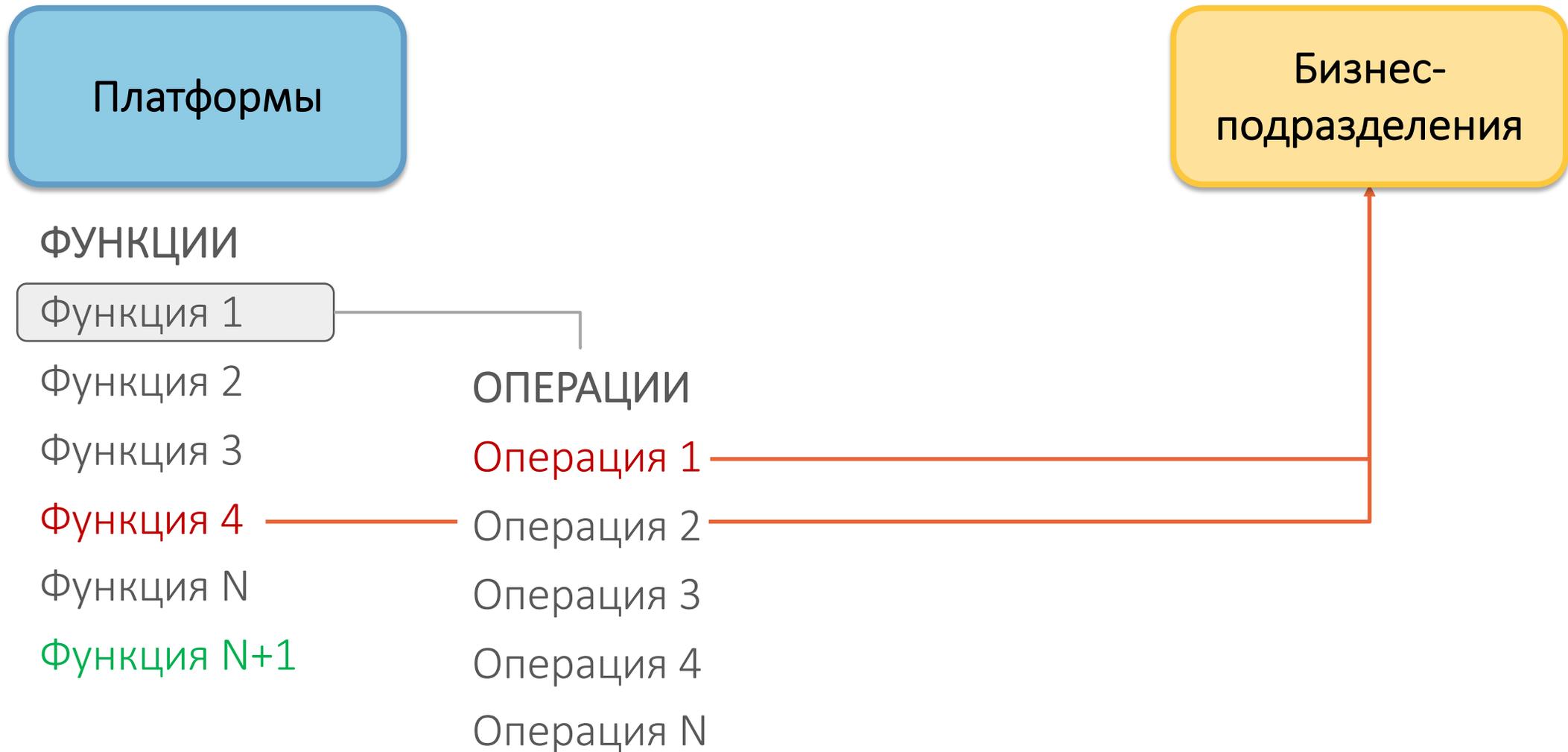
1. Определить, что является thinnest viable platform (TVP) для вас
2. Понять, что отделяет вас от этого состояния («лишний» функционал, не хватает API, etc.)
3. Положить в свой бэклог нужные задачи
4. Постепенно убрать все лишнее и приобрести все недостающее
5. НЕ СДЕЛАЙТЕ В ПРОЦЕССЕ ХУЖЕ



* Про другие типы расскажу как-нибудь потом



Как определить, что такое **thinnest viable platform**



Трансформация платформ началась! Но стало ли лучше?



НЕТ

Платформы

Current mood level



Бизнес-подразделения

Current mood level



**Определить модель
взаимодействия при дизайне
команды не менее важно,
чем её тип**

Модели взаимодействия команд

Одна команда предоставляет, а другая потребляет что-то «как услугу»

Stream-aligned команда 1

Stream-aligned команда 2

Stream-aligned команда 3

Платформа



X-as-a-Service



Модели взаимодействия команд

Одна команда помогает и наставляет (менторит) другую

Facilitating



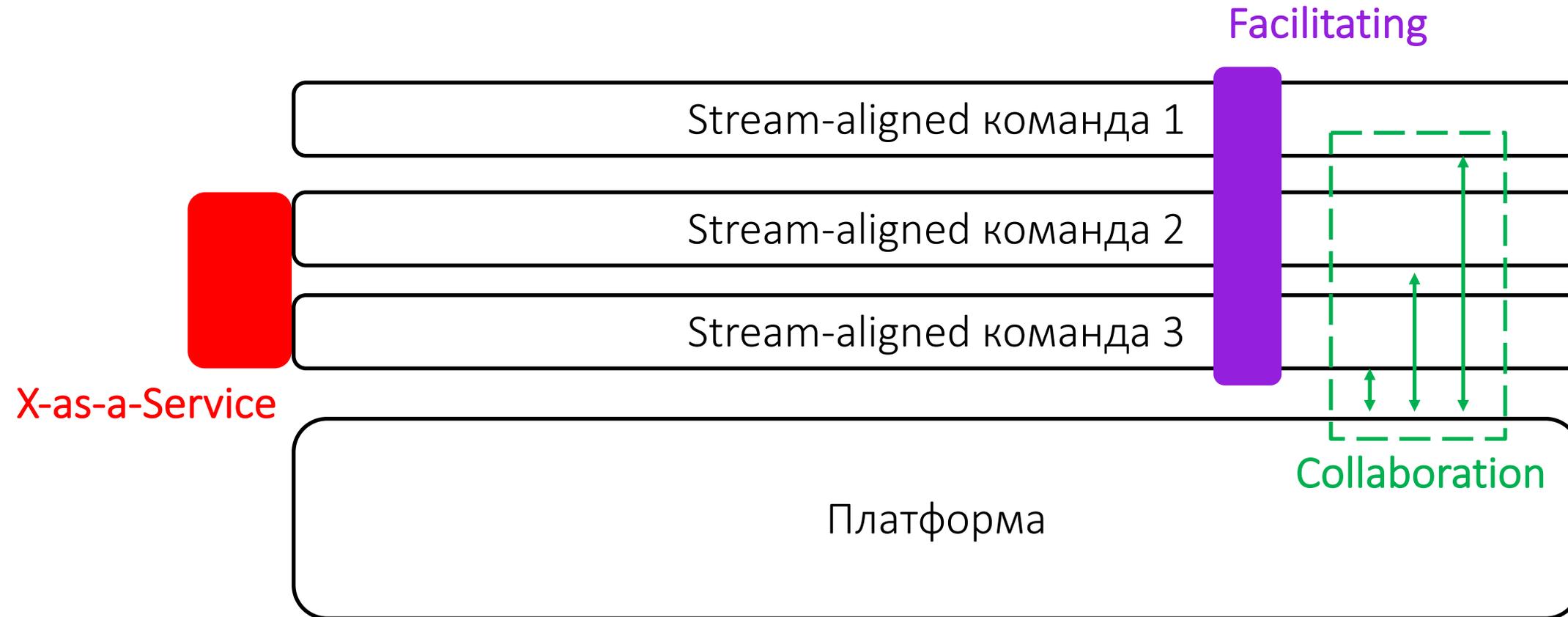
Модели взаимодействия команд

Команды работают вместе в течение определенного периода времени, чтобы делать новые штуки (API, практики, технологии и т. д.)



Модели взаимодействия команд

Все три типа взаимодействия нужны и важны



Как эту модель применить к продуктовым командам?

Collaboration

X-as-a-Service

Facilitating

Stream-aligned



Enabling

Complicated subsystem

Platform



Если Stream-aligned команда будет учить другие команды, у нее может не остаться времени на создание ценности для клиента!

А на платформенные?

Collaboration

X-as-a-Service

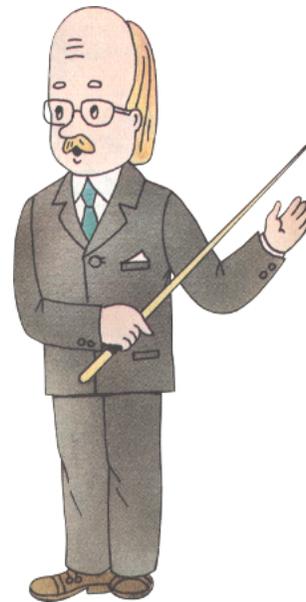
Facilitating

Stream-aligned

Enabling

Complicated subsystem

Platform



Если Platform команда будет кого-то учить, все будут ее хейтить!



Получается вот такая матрица

| | Collaboration | X-as-a-Service | Facilitating |
|-----------------------|---------------|----------------|--------------|
| Stream-aligned | | | |
| Enabling | | | |
| Complicated subsystem | | | |
| Platform | | | |

Платформы в ОТП сейчас



Core banking

~100 чел.



IT for IT

~20 чел.



Digital

~60 чел.



Платформы в ОТП сейчас

Current
mood level



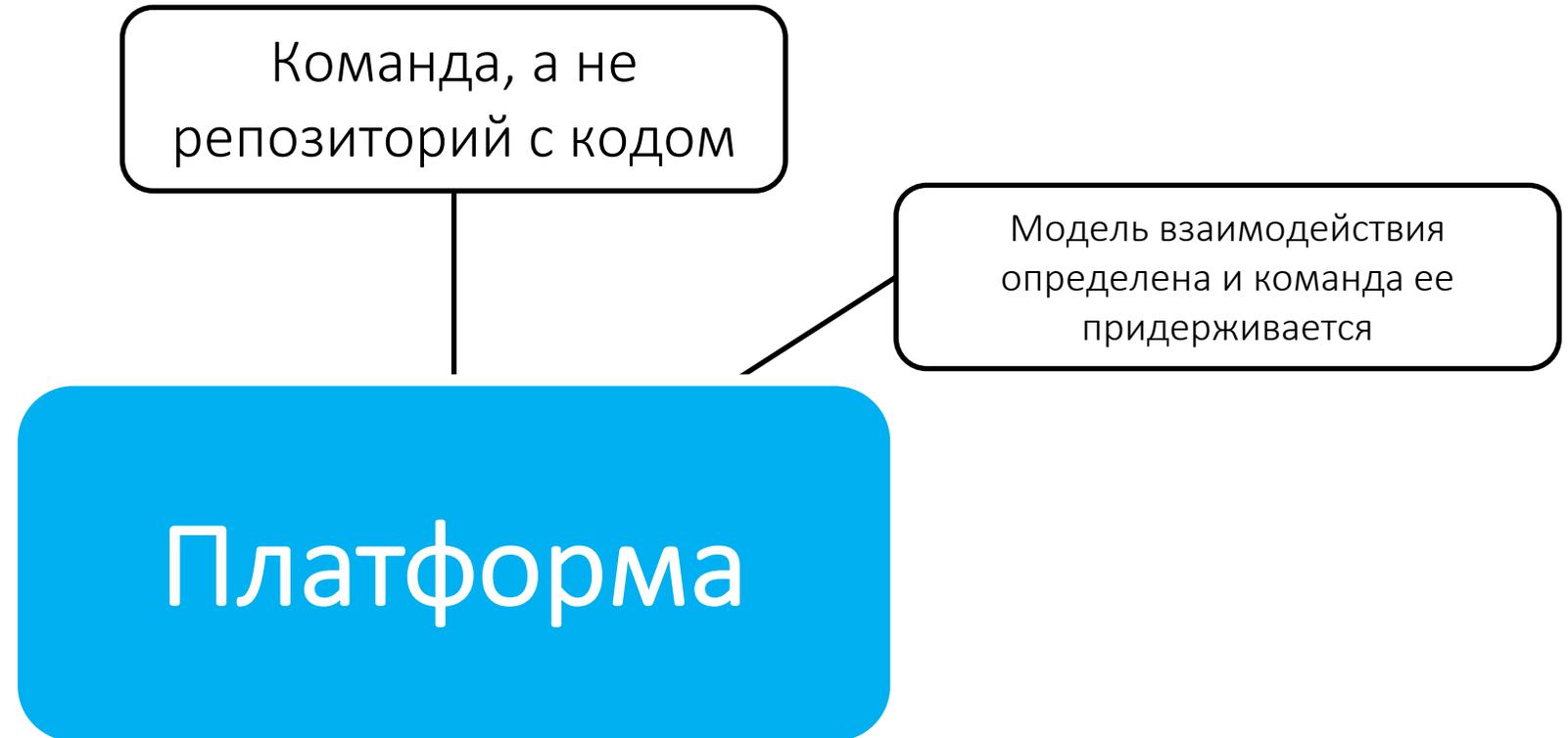
Definition of platform

Команда, а не
репозиторий с кодом

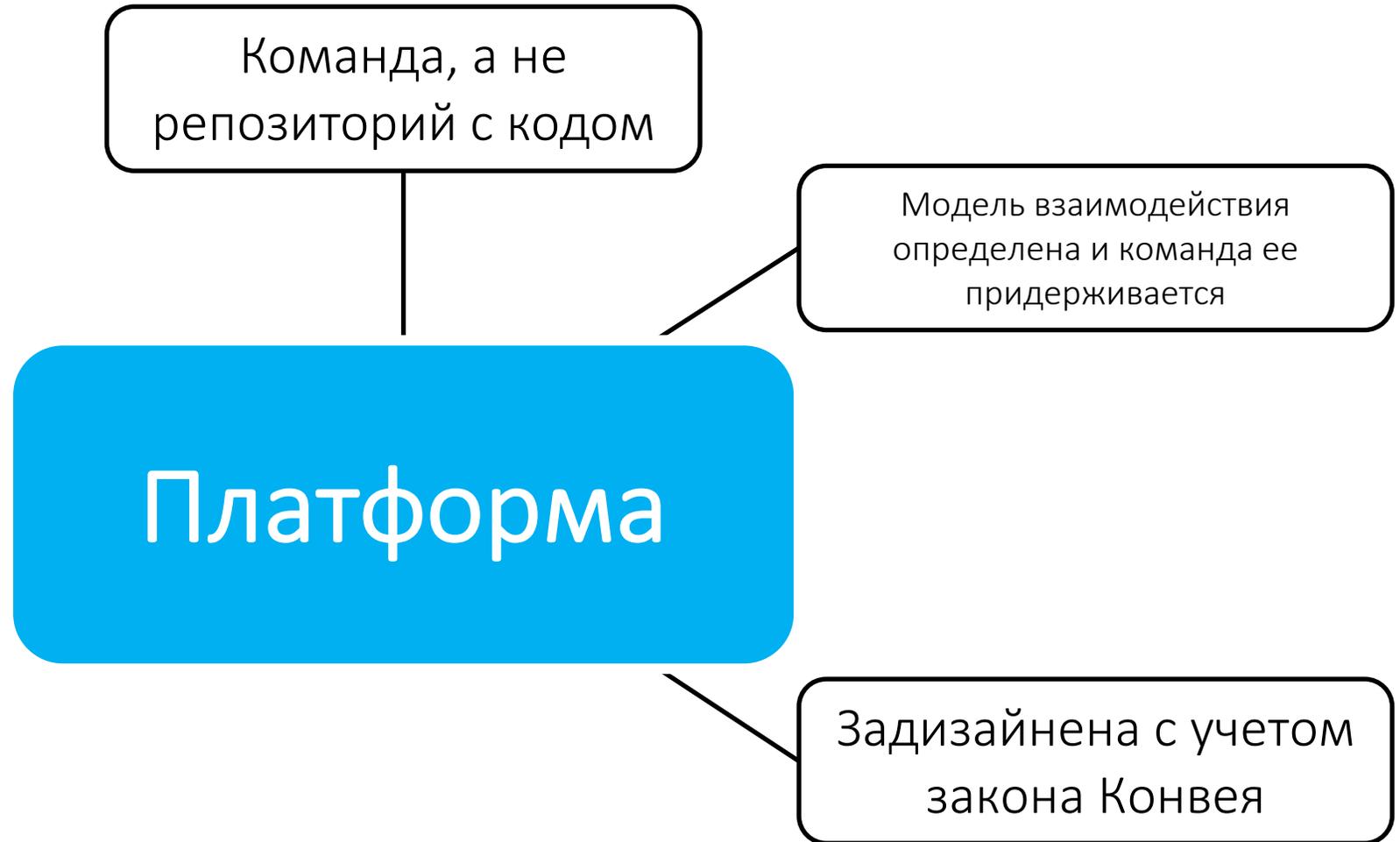
Платформа



Definition of platform



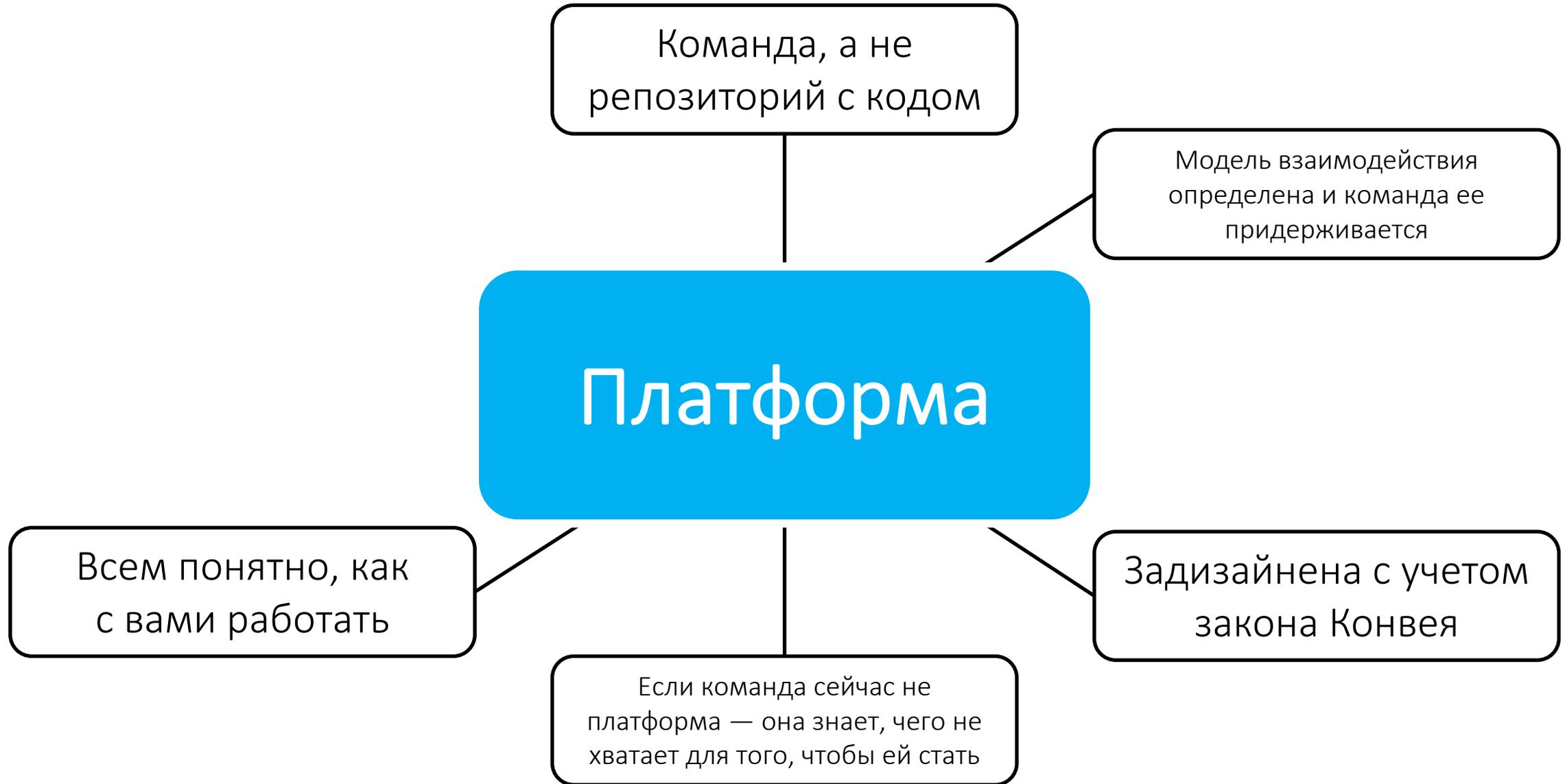
Definition of platform



Definition of platform



Definition of platform



Definition of platform



Выводы

- При формировании платформенных команд стоит опираться на четкую модель с типами команд и методами их взаимодействия

Выводы

- При формировании платформенных команд стоит опираться на четкую модель с типами команд и методами их взаимодействия
- Если эту модель не купят – люди будут уходить

Выводы

- При формировании платформенных команд стоит опираться на четкую модель с типами команд и методами их взаимодействия
- Если эту модель не купят – люди будут уходить (но это не всегда плохо)



Выводы

- При формировании платформенных команд стоит опираться на четкую модель с типами команд и методами их взаимодействия
- Если эту модель не купят – люди будут уходить (но это не всегда плохо)
- Настройка процессов— это не только code convention и общий CI/CD, но и оргструктурные изменения

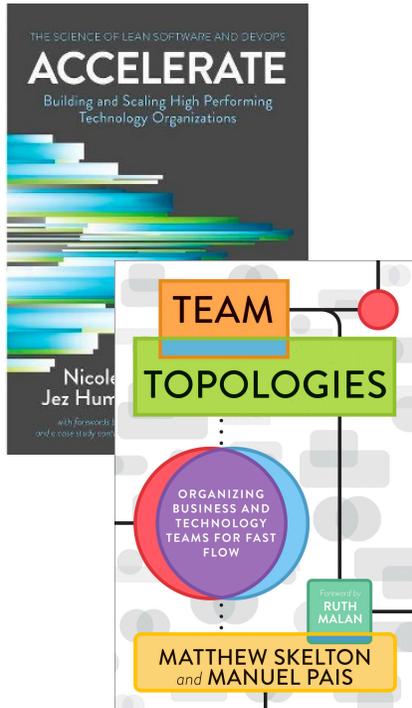


Выводы

- При формировании платформенных команд стоит опираться на четкую модель с типами команд и методами их взаимодействия
- Если эту модель не купят – люди будут уходить (но это не всегда плохо)
- Настройка процессов— это не только code convention и общий CI/CD, но и оргструктурные изменения
- Без вовлечения топов ничего не заведется

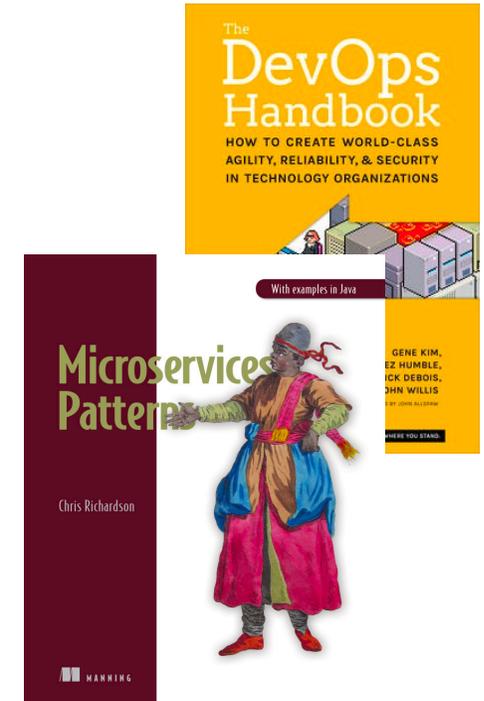


Нужно больше теории?



Организация
Продуктоцентричность,
автономные команды

Процесс
Lean/DevOps/CD



Архитектура
Микросервисы

Спасибо!



@m_bizhan



@mbizhan